# DOMAIN OF DEFINITION IN WARPER/MORPHER

Inventors
Dan Candela
Mitch Middler
Ron Brinkmann
George Wolberg

## BACKGROUND OF THE INVENTION

### Field of the Invention

[0001] The present invention relates generally to image processing using a computer system. In particular, the present invention is directed to a system and method for warping pixels from one location to another in a computer graphics environment.

### Description of the Related Art

[0002] In the image processing arts, there is often a need to alter an image such that it appears to transform over time into another image. This effect is commonly referred to as "warping". Given an image, an artist can create a source set of splines and a destination set of splines. As is known by those of skill in the art, a spline is a curve described by a point location, tangent length and tangent direction. A series of points around an object can follow the object quite closely, thus allowing it to be manipulated. Warping technology allows an image to be transformed from a first image to a second image by interpolating the pictures between the two splines.

[0003] For example, given a circle that is to be changed into a square, a spline around a circle is the source and around a square is the target. For each intermediate step between the two images, a warping algorithm calculates where the individual pixels within the frame should move to reach a final destination.

[0004] A morphing process is a specialized type of a warp in which the initial image and final image warp towards and dissolve into each other. As another example, to morph an adult face into a baby face, an artist might draw a set of splines around the adult's eyes, nose, mouth, ears, hairline, and jaw line; and then create corresponding sets of spline curves around the baby's features; the morph takes those pixels and changes their location from the adult's location to the baby's location, creating a scrunched-up face.

[0005] Alternatively, a warping operation can cause an image to grow – for example, if the baby's picture described above were warped into an adult face. However, if part of the image is moved outside of the visible window area, changes to the non-visible image are lost. In subsequent operations, if those areas are brought back into frame they are black and contain no image data. Conventional warper algorithms do not keep track of the image regions that are being warped. Consequently, if a warping or morphing process is applied to an image causing the image to move outside of the viewable window, it is no longer calculated as part of the image.

[0006] One conventional work-around to this problem has been to set the viewing area to be very large, in order to encompass any potential expansion of the image. However, this requires that enough resources such as RAM and processing power be available to render all of an image all of the time, even if it is not being displayed.

[0007] Accordingly, there is a need for a system and method for improved warping in an image processing and compositing environment.

## SUMMARY OF THE INVENTION

[0008] The present invention enables an efficient way to perform rendering operations in an image processing environment. A warping engine includes a warping module and a domain of definition (DOD) module. The DOD associated with an image is a preferably rectangular subregion of an image outside of which there is no non-background related information. When an object is warped, a new DOD for the image is determined by the warping engine by comparing the location of certain pixels in the warped image with their corresponding locations in the original image. By maintaining a DOD for the warped images, only that portion of the DOD that intersects with a region of interest is computed, thus resulting in optimized performance.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0009]** Fig. 1 is a block diagram illustrating a system in accordance with an embodiment of the present invention.

**[0010]** Fig. 2 and Fig. 3 illustrate the concept of a Domain of Definition (DOD).

**[0011]** Fig. 4 illustrates an image to be warped in accordance with an embodiment of the present invention.

**[0012]** Fig. 5 illustrates a warped image in accordance with an embodiment of the present invention.

**[0013]** Fig. 6 illustrates a warped image in accordance with an embodiment of the present invention.

**[0014]** Fig. 7 is a flow chart of a method for warping images in accordance with an embodiment of the present invention.

**[0015]** Fig. 8 illustrates a DOD determined according to an embodiment of the present invention.

**[0016]** Fig. 9 is a flow chart of a method for warping images in accordance with an embodiment of the present invention.

**[0017]** Fig. 10 illustrates a DOD determined according to an embodiment of the present invention.

**[0018]** The figures depict preferred embodiments of the present invention for purposes of illustration only. One skilled in the art will readily recognize from the following discussion that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the invention described herein.

## DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

**[0019]** Referring now to Fig. 1 there is shown a block diagram illustrating an embodiment of the present invention. As illustrated in Fig. 1, a warping engine **100** includes a warping module **102** and Domain of Definition (DOD) module **104**. Images to be warped are received by warping engine **100**. Warping module **102** includes logic to actually solve the warp, and DOD module **104** includes logic to determine a modified DOD applicable to the warped image. This functionality is described further below with respects to Figs. 2-10. Also shown in Fig. 1 are input image **106** and output image **108**.

**[0020]** Warping engine **100** employs Region of Interest (ROI) and Domain of Definition (DOD) paradigms to provide highly efficient image processing. The ROI of an image is a preferably rectangular region of the image that is of interest to the viewer. The DOD is a preferably rectangular region that defines the maximum boundaries of useful information in an image. Generally, for channels of the image located outside of the DOD, values will be 0. Unlike a region of interest, a DOD is preferably determined automatically. By limiting image processing based on the dimensions of the ROI and DOD, warping engine **100** can optimize processing and memory requirements.

**[0021]** For example, referring now to Fig. 2, there is shown an image **202** and a DOD **204** within the image. Also shown is a ROI **206**. In the illustrated example, the image **202** is a 2000 x 1500 pixel image, with a DOD of 500 x 500 pixels. Because the ROI is larger than the DOD in this case, all of the DOD is visible to a user. Suppose that the

DOD **204** is filled by a 500 x 500 pixel object that will be resized to be a 1500 x 1150 pixel image, which will be too big for the ROI—that is, the entire DOD will not be visible at once. As is known by those of skill in the art, it is preferable then not to expensively the entire 1500 x 1150 pixel object, but rather to calculate only the subset of the object that will remain within the ROI.

**[0022]** This is illustrated in Fig. 3, where DOD **304** has now grown to be a 1500 x 1150 region. ROI is still of the same size (1000 x 1000 pixels) as in Fig. 2, which means that the ROI is too small to display the entire image, and the part of the DOD outside of the ROI is therefore not computed.

**[0023]** As noted above, image warping is the repositioning of pixels based on transforms that modify the configuration of the image. Warping engine **100** provides warping functionality by allowing the user to specify some constraints that the warping solution must satisfy. For example the user may, by drawing shapes within the user interface, tell warping engine **100** to move a pixel that is currently located at position (100,100) in the image to location (200,200). Warping engine **100** then determines how to move all other unconstrained pixels within the image in order to produce a result. In this case, perhaps pixel (100,101) will be relocated to (200,201) based on the user defined input.

**[0024]** Fig. 4 illustrates an example of constraints including a source location **404** and destination location **402**. That is, the user has specified that the "W" in the word "Warp" should be warped from its original position, as indicated by the splines **404** to a new position, indicated by splines **402**. The result of the warp is illustrated in Fig. 5.

**[0025]** Using warping engine **100**, a user preferably has the option of having a default constraint placed on the image warp; the user can choose to have pixels along the edge of the image remain fixed or to move freely. In the warp illustrated in Fig. 4

and Fig. 5, the image border is fixed. However, if the same warp is applied to Fig. 4 without a border constraint, the resulting warp is different, as illustrated in Fig. 6.

**[0026]** In using free-form image warping as described above, it is non-trivial to determine a DOD, because there are an arbitrary number of constraints, often pulling the image in conflicting directions. The present invention uses two approaches, depending on whether pixels are to remain fixed at the image border, to accurately determine the DOD for such warps.

**[0027]** Referring now to Fig. 7, there is shown a flow chart of a method for determining a new DOD formed as a result of a warping operation in which pixels are fixed at the image border.

**[0028]** First, warping engine **100** examines **702** each of the user-defined constraints for pixel repositioning. By analyzing each constraint, warping engine **100** can keep track **704** of the largest displacement in each direction—up, down, left, and right—caused by the warp. After the largest displacement in each direction is determined, the displacement is then applied **706** to each of the corresponding edges of the original DOD rectangle to obtain a new DOD.

**[0029]** Fig. 8 provides an illustration of a DOD **802** obtained by using the method described above in conjunction with the image in Fig. 4.

**[0030]** In order to avoid wasting resources by unnecessarily rendering more of the image than will actually be viewed, the DOD of the image after warping should be determined. Fig. 9 illustrates a flow chart of a method for determining a new DOD formed as a result of a warping operation in which pixels are not fixed at the image border.

**[0031]** First, warping engine **100** determines **902** a forward warp resulting from the specified constraints. In a preferred embodiment, after warping engine **100** determines the forward warp there is a rectangular grid that specifies where each pixel in the image gets warped to.

**[0032]** For example, in one embodiment a mapping logically connects the following points:

(0, 0) goes to (10,10)
(1, 0) goes to (11, 10)
(2, 0) goes to (12, 10)

etc., where (*x*, *y*) refers to the pixel location at distance *x* from the left side of the image and *y* from the bottom of the image; (0, 0) being the lower left corner of the image.

**[0033]** To determine the new DOD, warping engine **100** examines the resulting pixel mapping grid at all points falling along the original rectangular DOD to determine **904** their new position after warping. For example, if the old DOD has lower left of (10, 10) and upper right of (100,100), then the top edge of the old DOD is a line from (10,100) to (100,100). By examining the new location of each pixel originally located on that line, warping engine **100** can determine **906** the top edge of the new DOD. For example, assume that the corner points at (10, 100) and (100, 100) did not move at all, but the pixel half way along the line at (55, 100) is relocated to (55, 120). By looking at every pixel along the old DOD top line, warping engine **100** determines that the maximum in the "up" (*y*) direction is now 120. Accordingly, warping engine **100** locates the top of the new DOD at position 120.

**[0034]** A similar analysis allows warping engine **100** to determine **906** the left, right and bottom edges of the new DOD. Thus, in a preferred embodiment, the maximum *x*, *y* of all of the points determine the upper right boundary of the new DOD and the minimum *x*, *y* determine the lower left.

[0035] In one embodiment, because computing a forward warp solution can be time consuming, warping engine **100** uses a low-resolution proxy image to determine the new DOD. Image dimensions and constraints are all scaled when using the low-resolution proxy.

[0036] Fig. 10 provides an illustration of a DOD **1002** obtained by using the method described above in conjunction with the image in Fig. 4.

[0037] The present invention has been described in particular detail with respect to a limited number of embodiments. Those of skill in the art will appreciate that the invention may additionally be practiced in other embodiments. First, the particular naming of the components, capitalization of terms, the attributes, data structures, or any other programming or structural aspect is not mandatory or significant, and the mechanisms that implement the invention or its features may have different names, formats, or protocols. Further, the system may be implemented via a combination of hardware and software, as described, or entirely in hardware elements. Also, the particular division of functionality between the various system components described herein is merely exemplary, and not mandatory; functions performed by a single system component may instead be performed by multiple components, and functions performed by multiple components may instead performed by a single component. For example, the particular functions of the warping module and so forth may be provided in many or one module.

[0038] Some portions of the above description present the feature of the present invention in terms of algorithms and symbolic representations of operations on information. These algorithmic descriptions and representations are the means used by those skilled in the image processing arts to most effectively convey the substance of their work to others skilled in the art. These operations, while described functionally or logically, are understood to be implemented by computer programs. Furthermore, it has also proven convenient at times, to refer to these arrangements of operations as modules or code devices, without loss of generality.

[0039] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the present discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0100] Certain aspects of the present invention include process steps and instructions described herein in the form of an algorithm. It should be noted that the process steps and instructions of the present invention could be embodied in software, firmware or hardware, and when embodied in software, could be downloaded to reside on and be operated from different platforms used by real time network operating systems.

[0101] The present invention also relates to an apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general-purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, application specific integrated circuits (ASICs), or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus. Furthermore, the computers referred to in the specification may include a single processor or may be architectures employing multiple processor designs for increased computing capability.

**[0102]** The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general-purpose systems may also be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will appear from the description above. In addition, the present invention is not described with reference to any particular programming language. It is appreciated that a variety of programming languages may be used to implement the teachings of the present invention as described herein, and any references to specific languages are provided for disclosure of enablement and best mode of the present invention.

**[0040]** Finally, it should be noted that the language used in the specification has been principally selected for readability and instructional purposes, and may not have been selected to delineate or circumscribe the inventive subject matter. Accordingly, the disclosure of the present invention is intended to be illustrative, but not limiting, of the scope of the invention.

**[0041]** We claim: